

Please type a plus sign (+) inside this box → ☒

# UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No.

TI-28385

First Named Inventor or Application Identifier

Jason M. Brewer

Title

System and Method for Loading Resolved Java  
Class Files to a Client Device

Express Mail Label No.

EL233151406US

On Page 1 of the specification, before line 1, insert -This application claims priority under  
35 USC § 119(e)(1) of provisional application number 60/117,550 filed 01/28/1999--

## APPLICATION ELEMENTS

See MPEP Chapter 600 concerning utility patent application contents

## ADDRESS TO:

Assistant Commissioner for Patents  
Box Patent Application  
Washington, DC 20531

1. ☒ \*Fee Transmittal Form (e.g., PTO/SB/17)  
(Submit an original, and a duplicate for fee processing)
2. ☒ Specification (preferred arrangement set forth below) [Total Pages **13**]  
 - Descriptive title of the Invention  
 - Cross References to Related Applications  
 - Statement Regarding Fed sponsored R&D  
 - Reference to Microfiche Appendix  
 - Background of the Invention  
 - Brief Summary of the Invention  
 - Brief Description of the Drawings (if filed)  
 - Detailed Description  
 - Claim(s)  
 - Abstract of the Disclosure
3. ☒ Drawing(s) (35 USC d113) [Total Sheets **2**]  
 a. ☒ Newly Executed (original or copy)  
 b. ☐ Copy from a prior application (37 CFR §1.63(d))  
 (for continuation/divisional with Box 17 completed)  
 [Note Box 5 below]
- i. ☐ DELETION OF INVENTOR(S)  
 Signed statement attached deleting inventor(s)  
 named in the prior application,  
 see 37 CFR §1.63(d)(2) and 1.33(b).
5. ☐ Incorporation By Reference (useable if Box 4b is checked)  
 The entire disclosure of the prior application, from which a copy of  
 the oath or declaration is supplied under Box 4b, is considered as  
 being part of the disclosure of the accompanying application and is  
 hereby incorporated by reference therein

6. ☐ Microfiche Computer Program (Appendix)
7. ☐ Nucleotide and/or Amino Acid Sequence Submission  
(If applicable, all necessary)  
 a. ☐ Computer Readable Copy  
 b. ☐ Paper Copy (identical to computer copy)  
 c. ☐ Statement verifying identical of above copies

## ACCOMPANYING APPLICATION PARTS

8. ☒ Assignment Papers (cover sheet & Documents(s))
9. ☐ 37 CFR §3.79(b) Statement (when there is an assignee) ☐ Power of Attorney
10. ☐ English Translation Document (if applicable)
11. ☐ Information Disclosure Statement (IDS)/PTO-1449 ☐ Copies of IDS Citations
12. ☐ Preliminary Amendment
13. ☒ Return Receipt Postcard (MPEP 503)  
(Should be specifically itemized)
14. ☐ Small Entity Statement(s) ☐ Statement filed in prior application  
(PTO/SB/09-12) Status still proper and desired
15. ☐ Certified Copy of Priority Document(s)  
if foreign priority is claimed
16. ☐ Other

\*A new statement is required to be entitled to pay small entity fees, except  
where one has been filed in a prior application and is being relied upon.

## 17. If a CONTINUING APPLICATION, check appropriate box and supply the requisite information below and in a preliminary amendment:

☐ Continuation☐ Divisional☐ Continuation-in-part (CIP)

of prior application No: /

Prior application Information: Examiner

Group / Art Unit:

## 18. CORRESPONDENCE ADDRESS



Customer Number or Bar Code Label

(Insert Customer No. or Attach bar code label here)



Correspondence address below

NAME	Carlton H. Hoel, Texas Instruments Incorporated		
ADDRESS	PO Box 655474, M/S 3999		
CITY	Dallas	STATE	TX
COUNTRY	USA	TELEPHONE	(972) 917-4365
		FAX	(972) 917-4418

Name (Print/Type)	Carlton H. Hoel	Registration No. (Attorney/Agent)	29,934
Signature		Date	1/25/00

Harden Hour Statement: This form is estimated to take 0.2 hours to complete. \*Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20531. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20531.

01/28/00

JCE75 U.S. PTO

09494213-012800



This application is a:: NONPROVISIONAL OF  
> Application One:: 60/117,550  
Filing Date:: 01-28-1999

Source:: PrintEFS Version 1.0

008210-87248460

SYSTEM AND METHOD FOR LOADING RESOLVED  
JAVA CLASS FILES TO A CLIENT DEVICE

5

FIELD ON INVENTION

This invention relates to loading resolved Java class files to a client device.

032350.A882

BACKGROUND OF THE INVENTION

5 Sending and receiving data over the Internet has become popular not only using personal computers and other conventional computer systems but also other types of devices such as cellular phones and personal communication devices and is even expected to be in television sets and set top boxes. These other type of devices have embedded microprocessors. Embedded devices are typically constrained by low memory and low power requirements. Java is a network oriented programming language associated with Sun Microsystems that is specifically designed for writing programs that can be safely downloaded to a computer via the Internet and immediately run. Java Virtual Machines are being placed on these embedded devices to interpret and execute Java applications, which consist of Java class files. Before a class file can be securely interpreted on a local device, it must be resolved and verified.

10 Fig. 1 illustrates the standard process for loading a Java application class. A Java application class 13 that resides on a network server 11 is downloaded to a client device 15. There the class file is verified by a security check and by determining that it is formatted correctly and loaded at load verifying device 17. A resolver 19 takes a class file (data) and creates a set of data structures that represent the fields and functions of the class that a specific interpreter 21 in Java VM on the client device can understand. This is a time consuming process that takes up computing resources and power. Furthermore, when a class is resolved, its representation in memory at the client

device 15 is expanded to fill out the data structures necessary for interpretation.

In order to reduce memory requirements for Java Development Kit (JDK) classes on Personal Java, Sun Microsystems has a tool called Java Code Compact that takes a set of class files and creates a binary representation of loaded and resolved classes. JDK is a software development package from Sun Microsystems that implements the basic set of tools needed to write, test and debug Java applications and applets. The Java Code compact process sometimes referred to as ROMizing a Java class through Sun's Java Code compact tool is shown in Fig. 2. A select set of classes 22,23 and 24 and all of their dependent classes (JDK Classes 2) are processed by the Java Code Compact tool 25, which preloads and prereresolves each class. The Java Code Compact tool 25 verifies and resolves the class files and creates a c-code representation of all the resolved data and the c-code is compiled into a binary image representation of the c-code. The output is loaded into read only memory sections 26 and read-write memory sections 27 containing the loaded class data. The binary object is then linked at linker 28 with the object code from the implementation of the Java virtual machine 29. During execution, a list of binary classes is created from these classes that have already been loaded and resolved. Any class not in this list must be dynamically loaded from some other source, verified and resolved before it can be interpreted.



DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram of prior art typical Java application load;

5 Fig. 2 illustrates Sun Microsystems' prior art class preloader with Java Code Compact;

Fig. 3 is a block diagram of one embodiment of the present invention; and

Fig 4 illustrates the method according to one embodiment of the present invention.



DESCRIPTION OF PREFERRED EMBODIMENTS  
OF THE PRESENT INVENTION

5 The present invention creates a binary representation  
of loaded and resolved classes at the gateway for  
dynamically loaded applications classes and then transfers  
the binary preloaded class to the client device rather than  
the application class file.

10 Referring to Fig. 3, there is illustrated one  
embodiment of the present invention. The server 31  
contains the application class 33. The application class,  
for example, is 4042 bytes. The client 35 loads the  
application through a gateway 37 at or wired to the server  
at the server location. The gateway 37 at the server  
15 includes a Java Code Compact or like device 39 that takes  
the class as described in connection with Fig. 2 and  
creates a binary representation of loaded and resolved  
classes. The Java Code Compact or like device 39 in  
gateway 37 retrieves the class over the wired network 41,  
20 verifies, preloads and preresolves the class to give, for  
example, a c-code representation of the preloaded and  
preresolved class. The gateway device 39 in applicant's  
system further includes element 39a that determines the new  
portion of the c-code representation. The element 39a at  
25 gateway 37 creates a binary representation of only the new  
portion of the c-code representation of the class. This  
binary representation of the c-code, along with memory  
location in the client direction provided at the gateway 37  
is sent over, for example, a wireless network 43, for  
30 example, to the client 45 having a binary class loader 47

and Interpreter 49. The binary class loader 47 takes and copies into the internal class structure in the Interpreter 49. The result is a full binary image that differs from the original preloaded and preresolved image by the addition of the application class. Since the client 45 already has the original classes that were preloaded and preresolved during the build process, only the new portion of the binary image with the new application class is needed. This portion is transferred from the gateway 37 to the client 45. Once on the client 45, the new preloaded class just needs to be copied into memory and added to the list of binary classes. This reduces the processing requirements on the client 45 from those needed to perform full verify and resolve process to a memory copy. The preloaded class can also be transferred faster since it is smaller than a regular class file. The sample values in Fig. 3 with an application class of 4042 bytes showed a size reduction to 2024 bytes (almost 50%).

The proposed invention allows applications to derive the same benefits of the preloader as the static set of devices classes resolved at build time. Since the classes are resolved and can be verified during this process, this reduces the computing requirements for this phase of interpretation. Less memory is also required since preloaded classes can eliminate redundancies that one found in fully resolved classes.

This lowers the bandwidth and time requirements to transfer application classes over a potentially slow wireless network 43 link between the gateway 37 and the

client 45. A further benefit can be obtained by caching these preloaded applications on the server. This would allow multiple clients to take advantage of an application that had recently been preloaded on the server.

5           Fig. 4 illustrates the steps of the method, wherein step 51, an Interpreter requests class to be loaded. In step 52, the gateway retrieves class over the network. In Step 53, the gateway creates a c-code representation of all data in the class. In step 54, it is determined where the new portion of c-code representation is. In step 55, gateway creates a binary representation of only the new portion of the binary representation of the c-code representation of the class. Step 56 sends only the new portion of the binary representation to the client via  
10           wireless network. In step 57, the binary Class loader in the client takes and copies into the internal class structure in the interpreter.

15

I CLAIM

1. A method for loading class files from a server to a client comprising:

loading an application class onto a gateway  
5 server that preloads and prereresolves said class;  
creating a binary representation of new portions  
of the preloaded and prereresolved class at said gateway; and  
sending only the new portion to the client.

2. A method for loading Java class files from a  
10 server to a client device comprising the steps of:

a. gateway retrieving a Java class file;  
b. gateway preloading and prereresolving said  
Java class file and creating a representation of the Java  
class file;

15 c. determining at the gateway new portions of  
said representation of the Java class file not loaded in  
said client device;

d. creating at the gateway a binary  
representation of only the new portion of said  
20 representation of the Java class file;

e. sending said binary representation of said  
new portion to the client device;

f. loading said binary representation of said  
new portion into said client device; and,

25 g. copying said binary representation into the  
internal class structures in the interpreter of Java  
virtual machine of the client device.

3. The method of Claim 2, wherein step b includes  
creating a c-code representation of the Java class file and

step c includes determining the new portions of said c-code representation, and step d creates a binary representation of only the new portion of said c-code representation.

4. The method of Claim 2, wherein said sending step  
5 e includes sending over a wireless network.

5. A system for loading Java class files to a client device comprising:

a. a gateway coupled to said server and responsive to a Java class file for creating a c-code  
10 representation of said class file;

b. said gateway creating a binary representation of said c-code representation;

c. a network coupled between said gateway and said client device for sending the binary representation  
15 to said client device;

d. a loader for loading said binary representation at said client device; and,

e. means for copying said binary representation into the internal class structure in an interpreter of said  
20 client device.

6. The system of Claim 5, wherein said gateway includes means for determining new portions of the said c-code representation, and in step b said gateway creates binary representations of only new portions of said c-code  
25 representations, and in step c said network sending only said new portions to said client device.

7. A method for loading Java class files to an embedded client device from a server comprising the steps of:

a. gateway retrieving a Java class file,  
b. gateway preloading and preressolving the Java  
class file to produce a representation of the Java class  
file;

5 c. determining at the gateway a new portion of  
the representation;

d. creating at the gateway a binary  
representation of only said new portion of the preloaded  
and preressolved representation of the Java class file;

10 e. sending said binary representation to the  
embedded client device;

f. loading said binary representation into said  
embedded client device; and,

15 g. copying said binary representation into the  
internal class structures in the interpreter of Java  
virtual machine of the embedded client device.

8. A system for loading Java class files from a  
server to an embedded client device comprising:

20 a. a preloader and preressolver in a gateway  
coupled to said server and responsive to a Java class file  
for preloading and preressolving a representation of said  
class file;

25 b. said gateway creating a binary  
representation of said preloaded and preressolved  
representation of said class file;

c. a wireless network coupled between said  
gateway and said embedded client device for sending the  
binary representation to said embedded device;

d. a loader for loading said binary representation at said embedded client device; and,

e. means for copying said binary representation into the internal class structure in an interpreter of said embedded client device.

9. The system of Claim 7, wherein said gateway includes means for determining new portions of said preloaded and preresolved representations of the class and sending only said new portions to said embedded client device.

ABSTRACT

5       The client loads an application class through a gateway server that preloads and preresolves a class, creates a binary representation of new portions of the preloaded and preresolved class, and sends only the new portion to the client.

05434218.012800



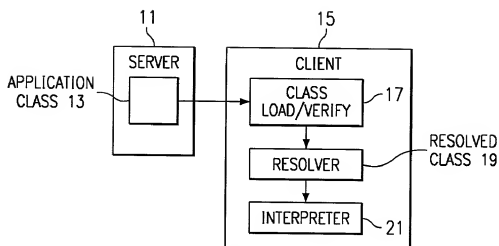


FIG. 1  
(PRIOR ART)

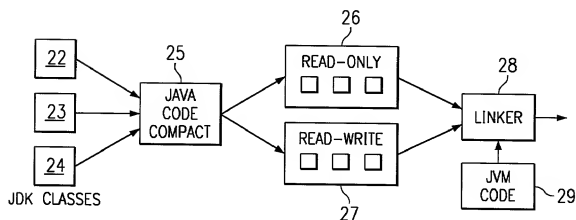


FIG. 2  
(PRIOR ART)

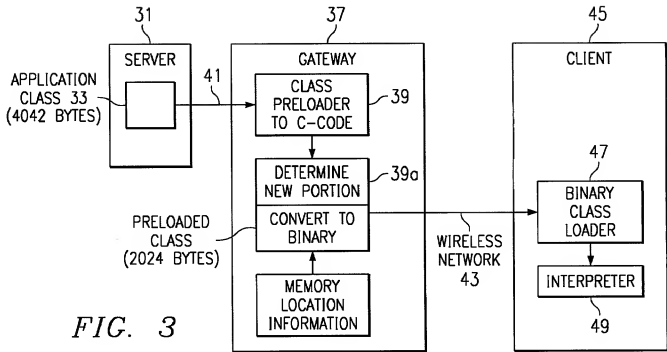


FIG. 3

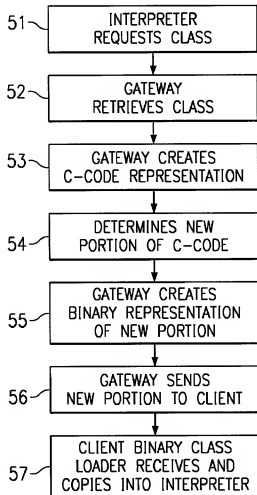


FIG. 4

00210-9126460

APPLICATION FOR UNITED STATES PATENT  
Declaration and Power of Attorney

As a below named inventor, I hereby declare that my residence, post office address and citizenship are as stated below next to my name; that I believe that I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought, on the invention entitled as set forth below, which is described in the attached specification; that I have reviewed and understand the contents of such specification, including the claims, as amended by any amendment specifically referred to in the oath or declaration; that no application for patent or inventor's certificate on this invention has been filed by me or my legal representatives or assigns in any country foreign to the United States of America; and that I acknowledge the duty to disclose to the U.S. Patent and Trademark Office all information known to me to be material to patentability as defined in 37 C.F.R. § 1.56.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

TITLE OF INVENTION:

SYSTEM AND METHOD FOR LOADING RESOLVED JAVA CLASS FILES TO  
A CLIENT DEVICE

I hereby claim the benefit under Title 35, United States Code, § 119(e) of United States provisional application USSN 60/117,550 filed January 28, 1999.

I hereby appoint the following attorneys to prosecute this application and transact all business in the Patent and Trademark Office connected therewith:

Robert L. Troike	Reg. No. 24,183
Robby T. Holland	Reg. No. 33,304
Carlton H. Hoel	Reg. No. 29,934
Richard L. Donaldson	Reg. No. 25,673
William B. Kempler	Reg. No. 28,228
Jay M. Cantor	Reg. No. 19,906
Robert D. Marshall	Reg. No. 28,527
C. Alan McClure	Reg. No. 31,041
Tammy L. Williams	Reg. No. 38,660
Ronald O. Neerings	Reg. No. 34,227
Charles A. Brill	Reg. No. 37,786

Please send correspondence to:

Robert L. Troike  
Baker & Botts, L.L.P.  
The Warner; Suite 1300  
1299 Pennsylvania Avenue, N.W.  
Washington, D.C. 20004-2400

and direct telephone calls to:

(202) 639-7710

Name of Inventor:

Jason M. Brewer

Residence & P.O.

5806 Goodwin Avenue  
Dallas, Texas 75206

Citizenship:

United States of America

Signature of Inventor:

\_\_\_\_\_

Date:

\_\_\_\_\_